**Домашнее Задание. Операторы**

Перед вами частичная реализация на C++ математической дроби. Класс Fraction из себя представляет структуру данных с приватными полями \_numerator (числитель) и \_denomenator (знаменатель). Публичный метод toString() - дает строковое представление дроби

Приватный метод reduce() - сокращает дробь, если возможно

Необходимо реализовать следующие операторы:

* Оператор +=, увеличивает данную дробь на целое число
* Оператор ==, сравнивает между собой две дроби
* Оператор +, складывает между собой две дроби

Для сокращения дроби при реализации программы можете использовать метод reduce, либо функции std::gcd (поиск НОД) и std::lcm (поиск НОК)\*

*\*Это функции из стандарта C++17. Убедитесь, что в настройках среды стоит стандарт не ниже C++17. Для Visual Studio:*

*Проект -> Свойства -> С/С++ -> Язык -> Стандарт языка с++ поставить с++17*

*Для VS Code - аналогично, или запускать в терминале командой:*

*g++ -std=c++17 main.cpp -o main*

#include <string>

#include <numeric>

#include <stdexcept>

#include <iostream>

class Fraction {

public:

Fraction(const int num = 0, const int den = 1);

Fraction(const Fraction &other) = default;

~Fraction() = default;

std::string toString() const;

Fraction& operator+=(int a);

friend bool operator==(const Fraction &left, const Fraction &rigth);

friend const Fraction operator+(const Fraction &left, const Fraction &rigth);

private:

void reduce();

private:

int \_numeratotr;

int \_denomenator;

};

Fraction::Fraction(const int num, const int den) {

if (den == 0) {

throw std::invalid\_argument("denomenator can't be zero");

}

\_numeratotr = num;

\_denomenator = den;

reduce();

}

std::string Fraction::toString() const {

if (\_numeratotr == 0) {

return "0";

}

if (\_denomenator == 1) {

return std::to\_string(\_numeratotr);

}

return std::to\_string(\_numeratotr) + "/" + std::to\_string(\_denomenator);

}

void Fraction::reduce() {

auto gcd = std::gcd(std::abs(\_numeratotr), \_denomenator);

if (gcd != 1) {

\_numeratotr /= gcd;

\_denomenator /= gcd;

}

}

Fraction& Fraction::operator+=(int a) {

// Напиши меня!

}

bool operator==(const Fraction &left, const Fraction &rigth) {

// Напиши меня!

}

const Fraction operator+(const Fraction &left, const Fraction &rigth) {

// Напиши меня!

}

int main() {

Fraction a(0, 10);

Fraction b(2, 5);

std::cout << a.toString() << " " << b.toString() << " " << (a == b) << "\n";

a += 2;

Fraction c(40, 20);

std::cout << a.toString() << " " << c.toString() << " " << (a == c) << "\n";

auto d = a + b;

std::cout << d.toString() << " " << "\n";

}